

PAL Show Scripting For Swissdeejays.com

A Set of Documentation and Tools

*Alexander Williams
exopilot@yahoo.com*

© 2005 by Alexander Williams.

All rights reserved. No part of this publication can be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers and/or authors.

While every precaution has been taken in the preparation of this text, the publisher assumes no responsibilities for errors or omissions, or for damages resulting from the use of information contained herein.

Introduction

In the SpacialAudio web forums, **swissdeejays** posted a request for help with certain facets of automation for their Internet broadcast station. There are multiple portions to the fulfillment of the requests they set, to wit:

- The creation of a system for managing the shift of play lists from day to day, to encompass House, Techno, and Trance.
- The creation of a PAL script that will interrupt the usual scheduling and insert a show at the appropriate place in the queue, with Intros and Outros.
- The creation of a PAL script and associated documentation for the insertion of a Station ID.

None of these issues is particularly difficult or overwhelming, but the delivery of an integrated solution is not a trivial task.

Phase #1: Daily Schedule Shifting

[As requested on the SpacialAudio Forums](#), Phase #1 of the request is the creation of an integrated method for shifting play list clockwheels on a daily schedule. To accomplish this, the following assumptions are made:

- There are already extant three Categories which are House, Techno, and Trance.
- The site is, in fact, running **SAM3**.

To accomplish this requires the creation of a Clockwheel for each day of the week. The easiest way to manage them is to simply name them as appropriate, ie. *Monday.cwf*. Place them in a directory together, named “SAM Clockwheels” or similar.

Each of the Clockwheels should contain the following:

Monday.cwf

```
Cat['House'].QueueBottom(smLemmingLogic, EnforceRules);
```

Tuesday.cwf

```
Cat['Trance'].QueueBottom(smLemmingLogic, EnforceRules);  
Cat['Techno'].QueueBottom(smLemmingLogic, EnforceRules);
```

Wednesday.cwf

```
Cat['House'].QueueBottom(smLemmingLogic, EnforceRules);
```

Thursday.cwf

```
Cat['Trance'].QueueBottom(smLemmingLogic, EnforceRules);  
Cat['Techno'].QueueBottom(smLemmingLogic, EnforceRules);
```

Friday.cwf

```
Cat['House'].QueueBottom(smLemmingLogic, EnforceRules);
```

Saturday.cwf

```
Cat['Techno'].QueueBottom(smLemmingLogic, EnforceRules);  
Cat['Trance'].QueueBottom(smLemmingLogic, EnforceRules);
```

Sunday.cwf

```
Cat['House'].QueueBottom(smLemmingLogic, EnforceRules);
```

Note carefully that the two-style days have two lines that will schedule the genres directly from the categories. If you want things a bit more random from those days, creation of a new play list containing all the tracks from both previous Categories should be created and an appropriate Cat[] call replace the two lines as in the single-category days' Clockwheels.

Once the Clockwheel files are in place, all that remains is to schedule the transitions on the **SAM3** Event Scheduler. For convenience sake, we'll assume all of the play list shifts occur at midnight on the day in question.

- Pull up **SAM3** and flip to whichever desktop contains the Event Scheduler.
- Enter the *Event name*. In this case, *Monday Clockwheel*.
- Select the Action *Load new clockwheel* and click on the folder icon to the right. Select your *Monday.cwf* file from the resulting dialogue.
- Click the *Scheduled times* tab.
- Select the circle in front of *Execute every*.
- Pull down *Monday* and make the time *12:00:00 AM*.
- Hit *+Add*.
- Click *OK*.

You've now added an event to change the Clockwheel to the Monday schedule at midnight on Sunday night. Repeat for each of the Clockwheel scripts already created, changing the day of week as appropriate.

Phase #1 is complete.

Phase #2: Scheduling Shows

For the individual special mix shows, there are, again, certain assumptions.

- Each of the shows has a play list file (*playlist.m3u*) stored on the server.
- Each show has a specific intro category (*MondayHouseIntro*) and an outro category (*MondayHouseOutro*) under one of the non-cycling Content folders, such as *Promos* or *Other*. In these intro and outro categories are the voice-overs desired for the show's play.

Effectively, the show script only has to manage a couple of things, primarily inserting the intro/show/outro in the queue and that's it.

ShowName.pal

```
{ShowName.pal
{Written by Alexander Williams}
{
  exopilot@yahoo.com
}
{Feb 19, 2005
}

PAL.LockExecution; {Run as fast as possible.}

{Since we'll be inserting at the top of the queue, insert the
necessary items in reverse order.}
Cat['ShowNameOutro'].QueueTop(smRandom, NoRules);
{Change DATASTORE to the directory where your show m3us are
stored.}
Queue.AddList('C:\DATASTORE\ShowName.m3u', ipTop);
Cat['ShowNameIntro'].QueueTop(smRandom, NoRules);

PAL.UnlockExecution; {Release the run loop.}
```

Once you have a ShowName.pal for each show, edited as appropriate for category names and directories, all you need to do is to create entries in the Event Scheduler for each of the shows. This process is identical to that found in **Phase #1** for scheduling a new Clockwheel, except instead of *Load new clockwheel* select *Execute PAL script* and select the appropriate show's corresponding PAL script.

If desired, the Intro or Outro can be substituted for a Station ID/Jingle in a straightforward manner by simply replacing the category name. Its suggested that if the station has a regular Station ID being injected regularly, the flow of programming seems to work better if you use show-specific intros and outros.

Updating Shows

Once the scripts are scheduled, they should require no updating. To update a show with new content, simply replace the m3u play list on disk with new files, named appropriately.

Automating this is left as an exercise for the reader, and depends wholly on how your DJs deliver their shows.

Phase #3: Jingle Scheduling

Configuring injected Station IDs and Promos differs from the scheduling of shows primarily in the frequency of their repetition. Using the Event Scheduler in such circumstances becomes unwieldy, because, in the current case, every hour of every day would require two events scheduled. Instead, we create a PAL script which is started once when the **SAM3** system comes online, then loops continuously.

The only assumption made for this **Phase** is that Station IDs are located in a category called *SwissdeejaysIDs*.

The Station ID injector script looks like this.

SwissdeejaysID.pal

```
{SwissdeejaysID.pal          }
{Written by Alexander Williams}
{      exopilot@yahoo.com      }
{Feb 19, 2005                }

{First, we instruct the script to loop forever.}
PAL.Loop := True;

{Then we tell it to run as quickly as possible.}
PAL.LockExecution;

{Insert the promo.}
Cat['SwissdeejaysIDs'].QueueTop(smRandom, NoRules);

{Unlock the loop to prep for ...}
PAL.UnlockExecution;

{... the time out wait.}
PAL.WaitForTime('+00:30:00');
```

Once this script is saved in your preferred PAL script directory, pull up **SAM3** and flip to the desktop with the *PAL Scripts* panel. Select *Automatically start script*, and then hit the folder and select *SwissdeejaysID.pal*.

Hit the small right-facing triangle on the top bar (*Play*) and it will start manually, insert an ID at the top of the queue, then begin waiting for 30min.

Finishing Touches

These scripts and procedures are, at present, relatively bare-bones. Enhancements could be made at multiple points, including having Station IDs not laid in during a given show or other, more complicated, scheduling arrangements. Hopefully, this will suit your current needs.